# Chapter 3
# Actor Bots

**Maria Arinbjarnar and Daniel Kudenko**

**Abstract** In many recent computer games there has been an increased use of Non-Player Characters as quest givers, tradesmen, members of the general public and as quest companions for the players. All these roles call for intelligent bots that can give the player an engaging experience by adjusting and adapting to the players' style of play. They need to behave as actors in a virtual improvisational theatre. They need to perform in a manner that the player finds believable and engaging so that the player becomes sufficiently immersed in the interactive drama and keeps coming back for more. The bot's intelligence needs to be both robust to respond to player actions and sufficiently complex to handle character traits, emotions and complex decision making that resembles that of a human player. A balance needs to be maintained between the bots' believability and their entertainment value for the player, as these two goals may be at odds in some situations.

## 3.1 Introduction

Modern technology offers us new and largely underutilized media in the form of vast virtual realities rich with graphics and sound. Large numbers of people log on every day to interact with others in online realities and some even run businesses selling virtual objects to other players. Additionally game worlds become ever richer in graphics and sound and players can spend months both in single player and online-mode; exploring, taking on quests and interacting with Non-Player Characters (NPCs). There are mainly three computer game genres that are of interest here;

M. Arinbjarnar (✉) · D. Kudenko
Department of Computer Science, University of York,
Deramore Lane, York YO10 5GH, UK
e-mail: maria@cs.york.ac.uk

D. Kudenko
e-mail: kudenko@cs.york.ac.uk

P. Hingston (ed.), *Believable Bots*, DOI: 10.1007/978-3-642-32323-2_3 

First Person Shooters (FPSs), Adventure games also known as point and click games and Role-Playing Games (RPGs).

FPSs games are centred on using a gun or any projectile weapon and with it killing a large number of enemies that are bent on killing the player. The games are thus very action based and the player needs good coordination and reflexes to master the challenges of the games. There is usually a main story line that runs through the game, and there are multiple levels that the player needs to complete to progress in the game.

Adventure games are quite the opposite of FPSs in that there is very little action or speed required to complete the game. This is why they are frequently called "point and click" games because for the most part all the player does is to locate or combine items with a click of the mouse or talk to NPCs in the game through drop down dialogue boxes to progress the story. They are very puzzle oriented and the player frequently needs to solve several puzzles to complete the game. These games tend to be heavily story centred and are frequently based on a film or a novel.

Computer-Based Role-Playing Games (CB-RPGs) are a combination of FPS and adventure games and Pen and Paper RPGs (PP-RPGs). In CB-RPGs the player takes on a role in a rather rich virtual world and through a series of quests will save this world from impending doom or resolve some really important in-game issue. The player will gain experience through solving quests and killing monsters and increase in power (level up) as the game progresses. These quests and levelling up combine elements from FPSs and Adventure games. There is considerable action needed to kill the monsters encountered and the player also needs to pick up objects and solve puzzles. CB-RPGs have a rich story element; there is a main story that needs to be played through to complete the game and multiple short stories in the quests that the player takes on. In PP-RPGs a group of players are needed to play the game using pens and papers to keep track of their character actions and experience gain. There is also a need for a Game Master (GM) who tells the story of the game, plays the NPCs and in general takes care of managing the game-play.

At the dawn of CB-RPGs, there where high hopes for the extension of PP-RPGs into computer games. There were visions of vast worlds with endless possibilities and rich interaction with NPCs, possibly meeting your favourite heroes from your favourite fantasy series. The player would not be limited by the imagination and storytelling skills of the GM or by how successfully a good group of players could be brought together. The players could adopt any role they desired, shape the character to their likings and play through a virtual reality, receiving a unique play experience for each character that they play. Included in the unique play experience would be a clear sense of a story and purpose for the characters actions just like a quality GM is able to create in PP-RPGs.

The main motivation for this vision is that it can be very difficult to find a qualified GM that offers sessions in the worlds and playing-style of an individual player's likings. It can also be very difficult to find a good group that fits the player's playing-style. Essentially what players are seeking in the CB-RPG is a RPG tailored to their tastes, competence and play style.

Current computer games fall short of providing a complete RPG experience for a number of reasons, the most prominent being that the story-lines are pre-authored and do little to accommodate an independent role-playing style. There are also pre-authored stories for PP-RPGs but the GM is free to diverge from them when it suits the game and GMs can also make their own stories either in advance or develop them on the fly during game play. Frequently the CB-RPGs force the player to violate the characteristic behaviours of the role in order to complete quests and progress in the game's main story. Additionally the NPCs' responses are pre-authored making them very repetitive. The results are rich game worlds that lack replayability due to repetitive story-lines and limited NPC interactions.

This is where actor bots come in. Actor bots behave in a similar manner as improvisational actors, acting out a drama and engaging the player in a dynamically changing storyline. The actor bots have three primary goals; staying in character for believability, working with other actor bots in creating a structured drama, and interacting with players in an engaging manner.

The chapter is centred around what part actor bots play in the transition of RPGs from being pen and paper based to being computer based and so we start by giving a comprehensive description of PP-RPGs, what the player expects and specifically what player types there are. Understanding different player types is very important to realise how actor bots can enhance player experience. In Sect. 3.3 we discuss how drama is playing an increasing role in current computer games and how the Directed Emergent Drama engine (DED) is designed to facilitate the merge of drama and games. We describe the DED engine and specifically the actor bots which are the main part of the engine. In Sect. 3.4 we discuss Bayesian Networks and how they are the core of the actors' knowledge base and decision mechanism. We also discuss problems and limitations of Bayesian Networks and evaluate their performance. In Sect. 3.5 we review related work and in the last section we summarise the chapter.

## 3.2 Players' Expectations

To be able to understand the motivation for actor bots it is very important to realise what many current CB-RPGs are trying to accomplish. They are trying to realise a rich game world that allows for very free and vicarious game play akin to a good PP-RPG. This is why understanding PP-RPGs is vital to understanding the aims of actor bots.

PP-RPGs are played with a GM and players typically sitting around a table and keeping track of the games progress using pen and paper, see Fig. 3.1. Dungeons & Dragons is the original form of the game that was introduced by Gary Gygax in 1974, published by TSR [43].

The GM is the narrator or storyteller of the game and will play the roles of any NPC that the players encounter in the game, such as shopkeepers and quest-givers. Additionally the GM will settle any dispute and has the last word when interpreting rules.

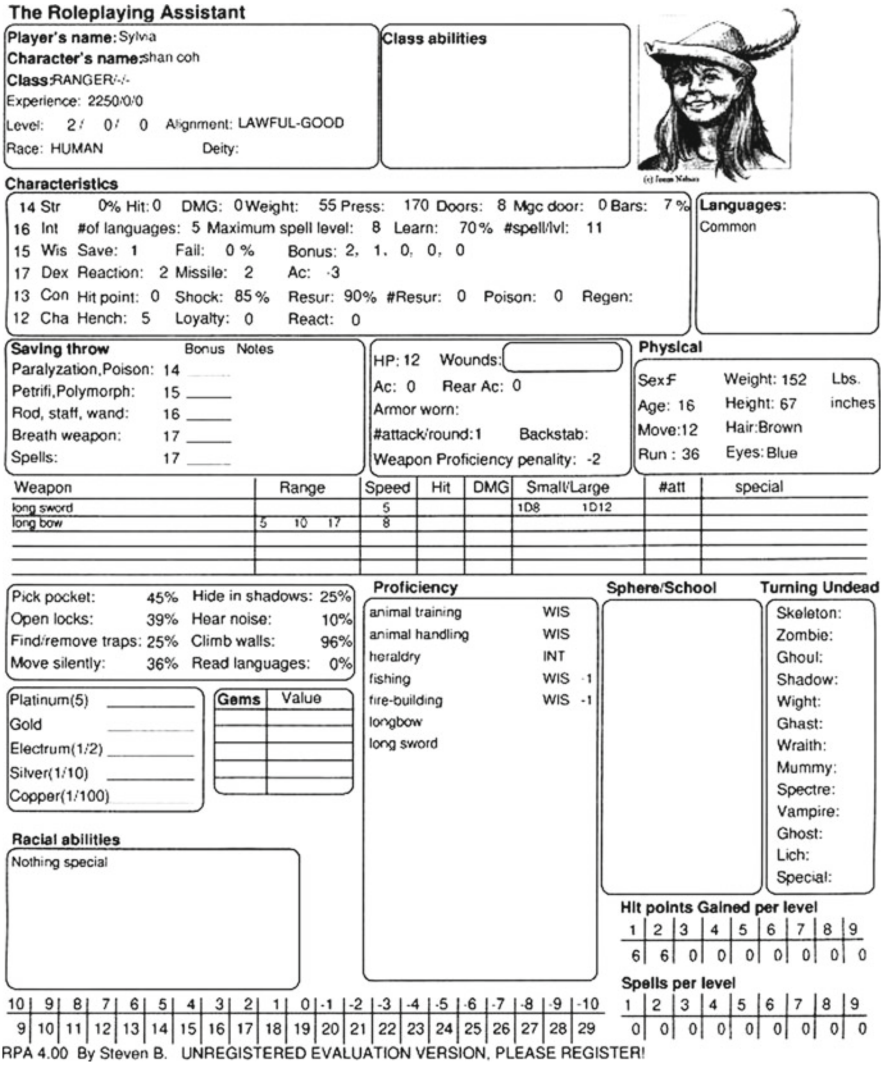
**Fig. 3.1** The basic setup for a PP-RPG

**Fig. 3.2** The many sided dices for PP-RPGs

The players create characters for the game and play a specific role in the game such as a wizard or a fighter. Each character has a set of statistics that define both physical make-up (e.g. strength, constitution) and mental abilities (e.g. intelligence, wisdom). Each character will have a detailed inventory and keep a record of which weapons and armour are equipped. The players keep track of this using pen and paper during the game. The characters have an alignment towards good or evil and chaotic or lawful.

The aim of the game is for the GM to create a rich and engaging adventurous world where the players can vicariously experience adventures by playing the role of their characters, taking on quests and exploring the secrets of the game world. The only limitation of the PP-RPG is that of the players' and the GM's imagination [17].

There are usually both large and small quests that the players can take on in addition to playing through a main plot that may possibly exist in the game-world. They will need to fight monsters and solve puzzles in order to complete the quests. They will be able to buy and sell items from merchants and rest in pubs along the way. When deciding the physical and mental limitations of characters and monsters



The Roleplaying Assistant

Player's name: Sylvia
Character's name: shan coh
Class: RANGER/-/-
Experience: 2250/0/0
Level: 2/ 0/ 0 Alignment: LAWFUL-GOOD
Race: HUMAN Deity:

Class abilities

Characteristics

14 Str 0% Hit: 0 DMG: 0 Weight: 55 Press: 170 Doors: 8 Mgc door: 0 Bars: 7 %
16 Int #of languages: 5 Maximum spell level: 8 Learn: 70 % #spell/lvl: 11
15 Wis Save: 1 Fail: 0 % Bonus: 2, 1, 0, 0, 0
17 Dex Reaction: 2 Missile: 2 Ac: -3
13 Con Hit point: 0 Shock: 85 % Resur: 90% #Resur: 0 Poison: 0 Regen:
12 Cha Hench: 5 Loyalty: 0 React: 0

Languages:
Common

| Saving throw | | Bonus | Notes |
|---|---|---|---|
| Paralyzation, Poison: | 14 | | |
| Petrifi, Polymorph: | 15 | | |
| Rod, staff, wand: | 16 | | |
| Breath weapon: | 17 | | |
| Spells: | 17 | | |

HP: 12 Wounds:
Ac: 0 Rear Ac: 0
Armor worn:
#attack/round: 1 Backstab:
Weapon Proficiency penality: -2

Physical
Sex: F Weight: 152 Lbs.
Age: 16 Height: 67 inches
Move: 12 Hair: Brown
Run: 36 Eyes: Blue

| Weapon | Range | Speed | Hit | DMG | Small/Large | #att | special |
|---|---|---|---|---|---|---|---|
| long sword | | 5 | | | 1D8 1D12 | | |
| long bow | 5 10 17 | 8 | | | | | |

Pick pocket: 45% Hide in shadows: 25%
Open locks: 39% Hear noise: 10%
Find/remove traps: 25% Climb walls: 96%
Move silently: 36% Read languages: 0%

Platinum(5)
Gold
Electrum(1/2)
Silver(1/10)
Copper(1/100)

| Gems | Value |
|---|---|
| | |

Proficiency
animal training WIS
animal handling WIS
heraldry INT
fishing WIS -1
fire-building WIS -1
longbow
long sword

Sphere/School

Turning Undead
Skeleton:
Zombie:
Ghoul:
Shadow:
Wight:
Ghast:
Wraith:
Mummy:
Spectre:
Vampire:
Ghost:
Lich:
Special:

Racial abilities
Nothing special

Hit points Gained per level

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |

Spells per level

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

RPA 4.00 By Steven B. UNREGISTERED EVALUATION VERSION, PLEASE REGISTER!

**Fig. 3.3** A character sheet for PP-RPGs

during game play the GM and players are aided by many sided dices and character sheets, as shown in Figs. 3.2 and 3.3.

Initially the games where set in a medieval world with castles, dungeons, elfs, goblins, wizards etc., i.e. a typical J. R. R. Tolkien type of a world. Since the dawn of RPGs in 1974 they have been adapted to many genres including mysteries (Etherscope [13]), sci-fi (Star Trek [10]) and vampire-tales. RPGs can potentially be adapted to any genre with a bit of imagination.

Anyone who has played and enjoyed a good PP-RPG knows how rich the experience can be, to feel as if you are that character treading slowly down a wooden path not knowing what monster can spring up in front of you or whether a group of travellers that can be seen in the distance are friend or foe. This element of surprise is in current CB-RPGs but only the first time you play it. While PP-RPGs can be potentially played for years and the novelty of the experience is only limited by the players' imagination.

Players actions are limited by rules that attempt with considerable success to emulate the natural physical constraints of the world. As Laurel [27] explains, it *"is difficult to imagine life, even a fantasy life, in the absence of any constraints at all"*, for example the time spent on travelling is not limiting a player's freedom, it is deepening the players vicarious experience.

> "Computer games are inherently limited because they only give you a set number of options. In a game like this (PP-RPG*), what we can do is limited only by our minds." Sam Weiss, as quoted in [42] * added by authors.

### *3.2.1 Player Types*

It is important to understand the different player types to realise player expectations and how actor bots can be of use. There are seven distinct player types that are described below, six of them are defined by Laws in [28]. Although it would be possible to designate other player types that grasp different aspects of the play-styles of players, it has not been done in a formalised way and Laws player types are describing the elements of play-styles that help us to identify what the Actor bots need to accomplish. Obviously the actor bots need to be catering to the needs or expectations of at least some of these player types or the exercise is futile. This is of greater importance than the game genre because the player types are more generic and applicable to all computer games. This does not mean that all computer games cater to every player type, quite the contrary. Moreover an individual can enjoy playing several different playing styles or a combination of two or more styles. It still remains that these player types are distinct and easily recognised. The interested reader can look at [48] for a more comprehensive discussion and user evaluation of the importance of player modelling.

As described by Laws there are six core player types; Power Gamer, Butt-Kicker, Tactician, Specialist, Method Actor, Storyteller and we add a seventh type that is frequently forgotten, the Casual Gamer:

1. **The Power Gamer**, loves to level up and acquire strong artefacts, magic and weapons. The current CB-RPGs cater well for this type of play it is mostly a matter of giving rewards with respect of the difficulty of the opponents fought and that is easy for a computer to calculate. This player is typically not very interested in the story but will appreciate companion bots that help him fight and to level up. If interacting with actor bots is embedded in the levelling up then this type of player will take care to play with the actor bots in order to level up

faster. When the Power Gamer has levelled up, the gamer will be searching for a more involved game and frequently joins a guild and or helps other players to level up. Clearly being able to participate in a deeper role appeals to this player type as currently when the main story is over or he has reached the highest level it is really game over for this player unless there is a good guild and other active players on-line.

2. **The Butt-Kicker**, tired after work or a stressful day, logs on as a strong character, for instance a fighter, and goes on a killing spree. Every First-Person Shooter (FPS) is designed for this playing-style. Still actor bots can come in as helpers and give a bit more depth to the game. That is what FPS games are heading towards. Most of their players want more depth to the game and they now have rather involved cut scenes that play out a kind of cinematic story for the player.
3. **The Tactician**, knows the rules, these players will plot and carefully calculate the movements of their characters, they like to be challenged. There are many strategy games on the market for this playing-style. As with the power gamer then this is mostly about calculations which computers are very good at. Many tacticians love to be in alliances and fight together with heroic comrades. A good alliance can be very hard to find.
4. **The Specialist**, usually has a favourite role and alignment that is always played, the player is constantly trying to improve on the role and really needs to be able to stay in character. This has little support in current CB-RPGs, although the players can pick roles and skills to advance. It needs more than that. A thief character would rarely venture out of cities where there are many opportunities for his trade in order to go on a goblin killing spree in a smelly dungeon. It does not really fit the role of a thief. Similarly a paladin will hardly care to sneak around somebody's dwellings breaking into locked containers to discover some vital information for the progression of a main story. Sadly this is repeatedly true for current computer games. Players find that they are unable to stay in-role if they want to complete the main story.
5. **The Method Actor**, is similar to the specialist except for not concentrating on a single role. The method actor tries to really enter the psyche of the character and to play the character in as a convincing way as possible. This has no more support than the specialist in current CB-RPGs. There is in fact no allowance for choosing specific characteristic except for charisma, intelligence, wisdom and alignment. There is no allowance for defining a caring, attentive cleric that becomes impulsive when angry, is slow to anger and bears grudges for a long time. Although players can of course imagine this when they play computer games then it does not have any actual affect on the game play as it does in PP-RPGs.
6. **The Storyteller**, this player enjoys a good story and will get quickly bored by needing to spend a lot of time levelling up as is required in CB-RPGs.
7. **The Casual Gamer**, this player is in for the ride and is frequently forgotten because of being passive. It is difficult to say whether one game or another specifically caters to their expectations. Still this is an important player type because it is one of the fastest growing type. The Casual Gamer frequently becomes a serious gamer when the player finds a good game. Moreover the Casual Gamer

is an essential fill in on-line games to show large numbers and to advertise the game by word of mouth. This is why a successful game should try to appeal to Casual Gamers as well as the serious gamers described above. Quite obviously, actor bots that engage players will help greatly in facilitating large numbers of Casual Gamers.

“Games give you a chance to excel, and if you’re playing in good company you don’t even mind if you lose because you had the enjoyment of the company during the course of the game." Gary Gygax, as quoted in [43]

What is mostly lacking in current CB-RPGs is to offer this companionship and the sense of a vicarious experienced based around the players personal play stile. Specifically for the Specialist, Method Actor and Storyteller but also for the other player types. This is where actor bots can play a crucial role in engaging the player and involving the player in a drama that is tailored to the player type and the players actions in the drama.

## 3.3 Drama and Games

If we compare the role of a GM to that of a CB-RPG we see that the CB-RPG is not really offering the same RPG experience. A CB-RPG has a pre-made storyworld which has only limited adaptability to players interactions. Many CB-RPG such as Fallout 3 [30] and Neverwinter Nights [6] try to incorporate such adaptability into their games.

Neverwinter Nights [6] uses alignment of the character to determine the available paths through the game. Each player creates a character and determines the initial alignment. The players get a list of options to use to interact with NPCs in the game; some of these options are for good alignment and some for evil. When players choose evil actions then their alignment is slightly shifted to evil and if they choose a good action the alignment is shifted towards good. Quests that are accepted and completed are also used to adjust players’ alignment. Good deeds adjust the players’ alignment to good and evil deeds towards evil. The alignment affects the player’s progress in the game by for instance hindering players playing druids in getting full access to their powers and the druid community if the player has not been careful to keep the character alignment as good. It will also affect what sub quests are available especially when it comes to quests that are for the advancement of a specific role and need a specific alignment. Still it does not change the main story or the main story plot, the player will need to finish specific main quests in order to advance and complete the main story. Nothing the player does will affect the pre-authored plots of the game.

Fallout 3 [30] has brought this idea even further, players actions greatly affect the characters reputation and opportunities. The availability of quests is directly linked to players actions; how much they explore the world, what they say in conversation, and what skills they pick when levelling up. Still the same can be said for Fallout 3 as with Neverwinter Nights: nothing the player does will change the pre-authored plots or the main storyline. The player is always playing along one branch of a multi-linear story that essentially follows the same fundamental plot-points each time. This means that players will not get a novel story experience when replaying the game, which makes it very repetitive and lacking replayability.

Another issue with these games is that players save and reload the game in order to get the plot development that they prefer. If they say something to an NPC that has negative results, they can simply reload an earlier saving and try again. Equally if they fail in a battle or a quest, they can simply reload an earlier save. This means that not only is the game trying to adapt to their play style but the players are also adjusting the outcome to their personal preference. This is a very natural behaviour on behalf of the player and there is nothing wrong with it, but it needs to be accounted for. Furthermore some players will systematically try every action and explore every area and put their acquired information on the web where other players can benefit from it if they choose.

This is not the case in PP-RPGs there is no save-reload function the story emergence depends on player actions and the GMs skill in providing an engaging and rich environment. The core difference is that PP-RBGs are interactive dramas while CB-RPGs are incorporating a narrative into a game. The latter has a number of problems that Juul relates well in [23]. Essentially you can't have narration and interactivity at the same time any more than you can have panorama and drama at the same time. The two are a contradiction. To narrate means to recount [37], when a player is interacting with a gameworld then it is at odds to try to tell the player a story at the same time. On the other hand the player can easily participate in a drama which at its core is a "play".

> "Pen-and-paper role-playing is live theatre and computer games are television." Gary Gygax, as quoted in [42]

As we have discussed, drama is clearly an integral part of RPGs despite the fact that some player types do not pay much attention to it such as the "Butt-kicker" and "Power gamer". Perhaps surprisingly drama is also becoming more and more integrated into First-Person Shooter (FPS) games.

FPS games are not very replayable in their basic single player form. The reason for this is that once you have played through it then when you play it again there are no surprises, no suspense. You know where the bosses are and how to deal with them; there are no significant changes between each game play and no new challenges. Multiplayer FPS are on the other hand highly replayable because then the player is constantly up against a new challenge. In recent years single player FPS games have evolved towards having the opponents, the Non-Player Characters (NPCs), more intelligent to provide a greater challenge for the player. They will simulate the normal behaviour of people in similar situations, including going to sleep, eating and in general going about their daily lives. One of the most recent and greatly popular games in these series is Assassin's Creed II [49] where the player is an assassin that is assigned a number of assassination jobs. In Assassin's creed as in many other recent FPS there is a much greater emphasis on the story and character development than before. The player gets an in-depth character description of the assassin and why he is in this job. The story progresses gradually through the game and is delivered in seamless cut scenes. The player is playing the protagonist and as the story progresses the protagonist faces many moral dilemmas and challenging revelations that dramatically change his perspectives and shape his character. This is very typically of other recent games such as Fallout 3 [30] and Heavy Rain [9]. Heavy Rain carries this character development even further and allows the player to make decisions on what the character does based on the characters emotions and thoughts, forcing the player to make challenging moral decisions. There is a very clear trend towards making the NPCs more intelligent and more autonomous in order to give the player a more fulfilling game experience. A good example of this is S.T.A.L.K.E.R. [50], there the NPCs actively simulate the behaviours of soldiers or mercenaries that try their best to kill the player in heavily polluted Chernobyl. Their behaviour is very believable and they provide challenging game play, they do not simply continuously walk the same circle. In this respect there is a clear disparity between the increased playability of FPS which has every potential of providing rich replayability, as becomes clear when these games offer multiplayer mode, and the still pre scripted unchanging storyline that the player is forced to follow. Creating more human like opponents is the topic of Chaps. 6 and 7, we concern ourselves with enriching the drama element.

The popularity of titles like Heavy Rain and Assassins creed clearly demonstrate that there is a market for games that provide richer story worlds and character development. This calls for an even greater emphasis on character development and drama that is not so pre-scripted that it is counterproductive to replayability. There needs to be a merge between drama and game-play. The drama should emerge around the player, influenced by the player actions and still provide a structured story experience. One way to accomplish this is to make the NPCs even more autonomous so that they can enact a drama for the player based on abstract goals rather than pre scripted story-lines. The NPCs need to be believable in their performance; this means that they should not violate the expectations of the audience. For example they need to stay in character such that a calm, gentile character remains calm and gentile unless extremely provoked etc. The NPC behaviour should be plausible in order to suspend disbelief. Effectively the actor bots should demonstrate human-like behaviour which calls for large knowledge bases and reasoning under uncertainty.

Bayesian Networks are specifically suitable for this task because they are causal and they provide a means of calculating action utilities based on probabilities in a highly efficient manner. Bayesian Networks are an effective way of incorporating psychological elements such as traits and emotions which are very important for believability [5, 7, 14, 19, 26, 47].

### *3.3.1 Directed Emergent Drama*

In order for the drama to become an integral part of game-play and support it rather than counter it we have designed an architecture called Directed Emergent Drama

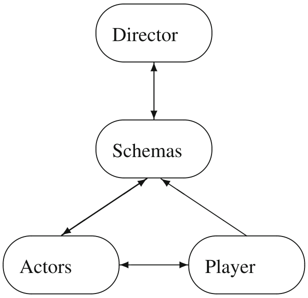



**Fig. 3.4** The DED architecture [1]

(DED) [1, 3]. As seen in Fig. 3.4 it consists of a director, schemas, actors and a player. The player only ever interacts with schemas and actors and never the director. The actors receive all their information from the player and from schemas, they do not interact directly with the director.

Rather than having predefined options like current CB-RPGs do, DED engine facilitates the player to act freely within the constraints of the virtual world. Actions for interacting with the characters are supplied by the schemas and are the same as those that the actors receive. These actions are multiple for each schema, players should not feel forced or railroaded in any way.

For example in a mystery drama players will frequently find themselves in an interrogation schema which provides all the core elements for the actors and players to participate in the schema. This includes a knowledgebase for the actors to understand what an interrogation is and what will be expected of their roles in that context. Additionally the actors receive any action (physical and speech) that is appropriate for that context, for instance asking about motive or means of the suspect. The actions received are generic and the actors and players need to fill it with knowledge from their own knowledge base. Generic action: *Ask about motive* may become: *Was Fred blackmailing you?*

### *3.3.2 Director*

The director overlooks the emergence of the drama and uses schemas to direct the drama by giving the actors appropriate schemas to play out, in this way the director is the GM in regards to directing the emergence of an engaging drama.

The drama can not move between acts until the objectives of the acts have been adequately satisfied. As an example, If a drama goal for Act I is to introduce characters, the drama will not move from Act I to Act II until characters' key traits have been exposed. If a character is to be intelligent, playful and curious then the character needs to have played out actions that are intelligent for a value above a given threshold $\mathscr{T}$, and the same for playfulness and curiosity.

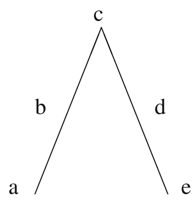



**Fig. 3.5** Freytag's Pyramid or "dramatic arc" [12]

Skilfully winning a chess game or making 3–4 correct estimates about, for instance, the age of old furniture or showing good arithmetic skills would be sufficient. The algorithm summarises the percentiles to see if it has reached the threshold $\mathscr{T}$. The director's role is to give the actors an opportunity to show their characteristics by choosing schemas that would be a good fit.

The director uses known structures as an aid in picking a suitable schema to develop an interesting drama. Freytag's Pyramid or "dramatic arc", [12], as shown in Fig. 3.5 is very useful. This referred to as a 'dramatic arc' [27] in this chapter. The dramatic arc outlines the rise and fall that can be found in an engaging narrative. The story will start with an inciting incident, which aims to capture the audience interest (a). Followed by a steadily climbing suspense, as the plot thickens, in order to further captivate the audience (b). Until the story reaches its climax (c). Followed by the resolution as the plot uncurls and the player learns the full truth (d) after which the drama may reach closure (e).

The director is not planning every detailed move of the actors, but rather the overall dramatic structure which makes his planning of acts and episodes a much more tractable problem. As with PP-RPGs it is the players and Actor bots that drive the story through their actions rather than the Director or GM.

### *3.3.3 Schemas*

Each schema has a finite set of roles that are annotated as being essential or non-essential. It is only necessary to fill the essential roles to successfully execute a schema; the non-essential roles add variety and increase flexibility. Each role is annotated with a finite set of characteristics that it supports. The characteristics also have a numerical value attached to them, this represents to what degree the display of this characteristic is supported by the role.

The director uses the set of characteristics to match the roles to actors, deploying the schemas that best compliment the various characteristics of the characters. The director is not in a good position to make decisions about direct interactions with the player, because the director would need to be constantly aware of everything that takes place—including the internal state of every character in the drama. This would quite rapidly escalate into an intractable computation problem for the director.

### 3.3.4 Actors

Actors play characters in the drama, their main task is to decide on an appropriate action if any. There are three primary conditions for acting; *response to stimuli, response to internal process, response to a request to act made by other actors.*

#### 3.3.4.1 Response to Stimuli

When an actor or a player acts, it will inform each of the actors that share the same scene of the action taken. When an actor is informed of an action then a separate internal thread first checks whether the action is directed at the actor and whether the action is detailed in any of the drama-schemas that the actor is currently in. If neither applies then the action is not relevant to the actor and the actor will not attempt a response, the actor will add any gained knowledge from the action to its knowledge base, (for instance information regarding the fulfilment of schema goals and info on the current location of other actors).

If the action is relevant to this actor then the actor will start a separate thread that evaluates an optimal response.

The algorithm is as follows:

```
For actor a1 and actor a2.
—a2 Sends out a notification to all actors on the scene that it has
taken an action.
—a1 Reads the notification and realises that it was a
question addressed to it.
—a1 Computes a set of optimal responses to the question with respect
to its knowledge base, character,
situation, emotions and its current goals.
—a1 adds the set of optimal responses to an array of
other applicable actions with added weight to indicate
priority.
—a1 then evaluates what the set of optimal actions is
and picks one to execute.
```

As seen in the algorithm, the actor needs to first add his set of optimal responses to an array containing other applicable actions because the actor may be preoccupied with something else. For example the actor may be talking on the telephone and not ready to answer.

#### 3.3.4.2 Response to Internal Process

The actor will also act due to an internal process that initiates an action. This is to engage the player if the player has been inactive or if there are unfulfilled drama goals that will progress the drama but the player has not initiated actions that would lead to them. All actors have both drama related goals and character related goals. The drama related goals are of the form of aiding the drama progress as expected. For instance, actors have the goal of revealing relevant clues to the user. The actors also have character related goals such as showing specific characteristics or hiding their lack of alibi in order to portray a believable character.

#### 3.3.4.3 Response to Request

During the unfolding of the drama each actor is responsible for both acting a convincing role and to fulfilling drama-specific schema goals [2]. For instance, in an interrogation where two characters John and David are present and David has a very strong false alibi then the actor bot playing David can make a request of the actor playing John that he has John reveal the flaw in the alibi as David would hardly sabotage his own cover directly. John might then say "How could you have caught the 08:15 train when I saw you at 08:20 in the garden". This does not require advanced logistics. The Bayesian networks facilitates this type of sentence generation so long as it is domain specific and the Bayesian networks facilitates such domain specific querying due to the fact that the BN is causally structured. The actor playing John searches for any knowledge in the specific domain that will increase the utility of the opportunity variable.

The algorithm is as follows:

Where David's actor is $a_1$ and John's actor is $a_2$.
−$a_1$ recognises that he needs to reveal that his alibi is faulty.
−$a_1$ recognises that he can't reveal that directly to the user in a believable way.
−$a_1$ sends a request to $a_2$ in the form of drama-goal: reveal opportunity of $a_1$.
−$a_2$ queries his knowledge base for actions that will satisfy the goal.
−$a_2$ adds any action that satisfies the goal in the request to an array of possible actions with added weight to indicate priority.
−$a_2$ then evaluates what the set of optimal actions is and picks one to execute.

As seen in the algorithm, $a_2$ will not necessarily act on the request. $a_2$ will first query its knowledge base for relevant actions and after adding them, with suitable weights, to the set of other relevant actions, $a_2$ will pick an action from the whole set

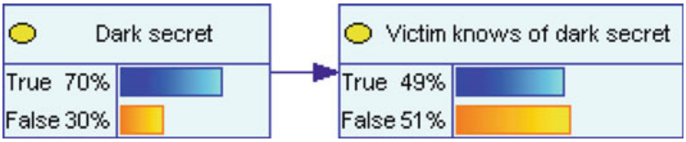



**Fig. 3.6** A simple Bayesian net

of possible actions based on $a_2$'s characteristics and other goals. This means that $a_2$ may or may not aid the requesting actor depending on context and $a_2$'s priorities. This is necessary as it may not fit either $a_2$'s knowledge base or $a_2$'s current interaction with the user to comply with $a_1$'s request.

## 3.4 Bayesian Networks

Bayesian Networks (BNs) are very good to reason about uncertainty such as in human behaviour because they provide the means to determine the probability of the various possible causal outcomes. The variables in a BN can represent anything ranging from system mechanics (e.g. whether there is fuel in a car or how fast the car can accelerate) to the intricacies of human traits and emotions. BNs offer the possibility of mapping the transiency of emotions and the resulting transient decisions that humans make. There has been a growing interest within Psychology and Philosophy as well as in Computer Science on using BNs to model human thinking [14, 26, 47]. We use causality extensively in our daily lives from deciding how to best manage to get the kids to school on time, to trouble shooting why the car will not start, and also when deciding on actions when highly influenced by emotions like anger or fear.

There are obviously many other AI techniques that have been used to try to emulate or simulate human like thinking, e.g. planning, search, neural networks etc. No AI method, to date, has been entirely successful. In fact all of them still have a long way to go to reach any sort of consistent believability for any length of time with minimum of complexity in the agents' environment. Since we are specifically making believable actor bots for drama based computer games, we are not really interested in fully emulating human like thinking but rather to give an appearance of human like reasoning. We discuss other approaches to achieve similar results in Sect. 3.5.

A BN is a directed acyclic graph made up of variables and arrows between the variables that indicate inference [22]. In Fig. 3.6 a simple Bayesian net is shown with two variables {*dark secret, victim knows of dark secret*}. These variables are found in the characters' knowledge bases and the former indicates whether a character has a dark secret or not and the latter indicates whether the victim knew of it, which could indicate the possibility that the victim was blackmailing the character which is a motive for murder. Each variable has two or more states that are mutually exclusive,

**Table 3.1** The definition of the *victim knows of dark secret*

| Victim knows of dark secret | Variable | |
|---|---|---|
| Dark secret | True | False |
| True | 0.7 | 0 |
| False | 0.3 | 1 |

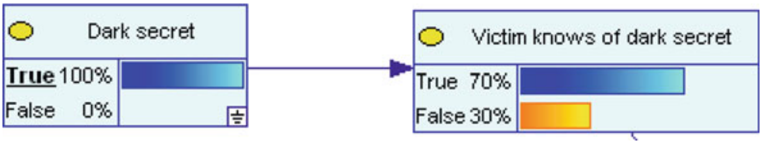



**Fig. 3.7** Dark secret instantiated to true

in Fig. 3.6 both variables have the states {True, False}. The directed arrow indicates a causality between *dark secret* and *victim knows of dark secret*. The victim cannot know of any dark secret unless there is one.

The percentages shown in Fig. 3.6 are inferred values from calculating the network. There are underlying values that are defined in the network before calculating it. First we look at the *dark secret* variable which has no parent variable. Its defined values are simply {True = 0.7, False = 0.3}. The *victim knows of dark secret* variable is slightly more complex because it has *dark secret* as a parent, each state of the parent needs to be assigned a value, see Table 3.1

The percentage values that are assigned to each state are the inferred values from the net. For example, the value of {True} of the *victim knows of dark secret* variable will increase if the value of {True} of the *dark secret* variable increases. If *dark secret* state {True} is set to equal one (instantiated) then the inferred values change as can be seen in Fig. 3.7.

### *3.4.1 The BN Architecture*

The variables in the actor bots BNs represent knowledge items such as hair colour, shoe size, relationship status, scenes and objects in the virtual world and the causal connections between these. Each state of a variable can be used to make a speech action. For example, in Fig. 3.8, the variable shoe size with states in the range 36–45 can be used by the actor bot to say that Kenneth's shoe size is 42. For example, the actor bot has been given evidence about basic attributes such as shoe size. This means that the actor bot has evidence for state 42 of the shoe size variable, see Fig. 3.8.

The actor bot can use this to form the speech "Kenneth's shoe size is 42" by mapping the state of the variable to the corresponding authored sentence, e.g. "{actor bot} shoe size is {value}" = Kenneth's shoe size is 42. If the actor bot is not certain, as in Fig. 3.8, the sentence can be Kenneth's shoe size *could be* 42. We format sentences

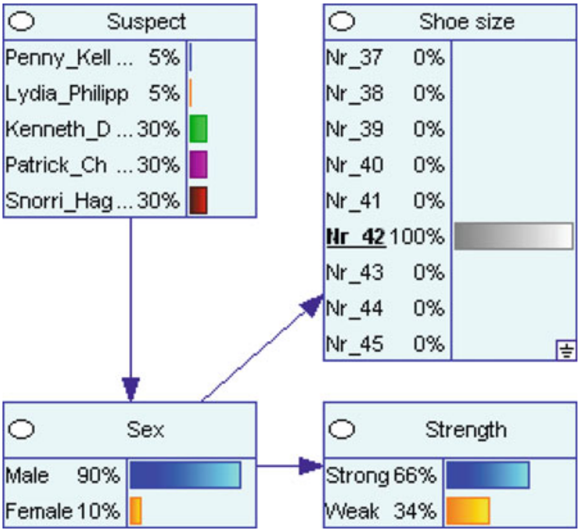



**Fig. 3.8** A small example to demonstrate sentence creation

in this way by authoring them in XML and attaching tags that indicate what the input and goal variable is. In this example the input variable is shoe size and the goal is suspect.

The example in Fig. 3.8 is just a small piece of the whole net. We use Object Oriented BNs (OOBNs) [24] to create an actor bot's network from a script file containing descriptions of around 150 variables. These variables describe the bare essentials of a mystery plot such as the motive, means, opportunity and character features. From these variables a much larger network is created to represent the actor bot's beliefs of itself and of other actor bots in the drama. The actor bot also has beliefs of what each of the other actor bot's beliefs are of each other. The resulting BN has over 3,000 variables.

## *3.4.2 Complexity*

Three thousand variables is not considered to be a very big network to implement a virtual actor bot. We expect that an average actor bot in an interactive drama could easily have a few hundreds of thousand variables. However, even this small BN takes several seconds to update with basic BN reasoning algorithms, which clearly is not sufficiently fast for real-time applications. Not only is it not fast enough but it also leaves no room for all the other processes, such as graphics and animation that are highly demanding on resources in any high graphics computer game.

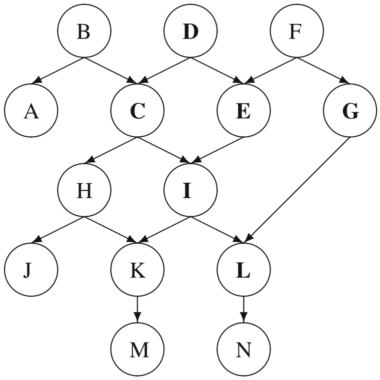



**Fig. 3.9** A small Bayesian net to explain relevance reasoning

### *3.4.3 Relevance Reasoning*

Relevance Reasoning is a well-known technique for BNs, that uses d-separation to determine which variables need to be updated to compute inference on one or more target variables given evidence [29]. D-separation holds when two distinct variables A and B in a network are not affected by each other receiving evidence. In other words, A can be given evidence without it affecting the inference of B and vice versa [22]. Relevance Reasoning identifies the variables that need to be updated based on which variables receive evidence and which variables become affected and are needed to correctly compute the inference of the target variables. This method greatly speeds up the updating of the network in the general case. The worst case is if all the variables in the network are affected by the evidence and are needed to compute the target variable. This means that in the worst case it is NP-hard.

For example if we look at the small BN in Fig. 3.9. If the input variable $I$ is **C** and the target variable $T$ is **L** then only variables C, D, E, G, I and L need to be updated to get correct values in **L**.

### *3.4.4 The Process*

When an actor bot needs to decide on a speech action, the actor bot will take the following steps:

1. Find applicable sentences
2. Evaluate each sentence
3. Choose an optimal sentence

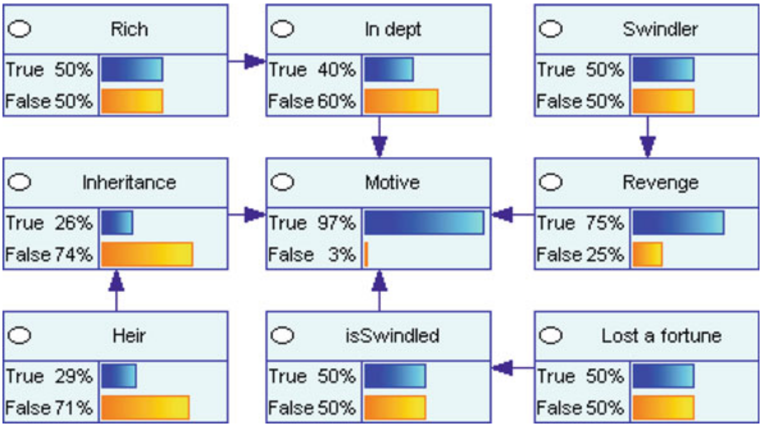



**Fig. 3.10** A small example to demonstrate context

#### 3.4.4.1 Applicable Sentences

First the actor bot finds a set of applicable sentences to evaluate, by querying the BN for a set of sentences that satisfy the goal and are contextual to the input. The input is any speech act that is causally connected to what was last said in the conversation. For example in Fig. 3.10 then all the variables are contextual to the $Motive$ variable. If any of their states is given value, it will affect the belief of the $Motive$ variable. For example if the $Lost\ a\ fortune$ variable is instantiated to true, it will positively effect the belief of the $IsSwindled$ variable, increasing the probability that the suspect was being swindled by the victim.

The actor bots first decides which actions are applicable. This the actor bot does by means of relevance reasoning as described above. $T$ (target) is the goal of the actor bot and $I$ (input) is for instance a speech action that the actor bot is evaluating to realise a goal. The actor bot uses $T$ and $I$ to extract the corresponding object $O$. $O$ is extracted from a fully updated BN with the values that the variables in $O$ had when in the BN. Because the BN is fully updated then the values in each variable are valid and up to date. The input values are then given as evidence to $I$ and the values are updated for all the variables in $O$. The values of $T$ are then read and their distance (difference between the intended values of the goal and the inferred values of the goal variable) from the goal is stored. Doing this repeatedly for all the actions that the actor bot is evaluating, results in a set of applicable actions. Using the net in Fig. 3.9 as an example, if the C variable is $I$ and L is $T$ then we extract (C, D, E, G, I, L) as $O$. C then receives evidence and only the variables in $O$ are updated. The result is the values in the L variable.

For example, see Fig. 3.10, if we want a sentence that says that *motive* is true $= 100\,\%$ and if we put evidence in the *Lost a fortune* variable, true $= 100\,\%$. The sentence could be: "Kenneth could have a motive because he lost a fortune". The *motive* variable would then have true $= 98\,\%$ which makes the difference between the target 100 % and the value 98 % be 2 which is stored with the possible sentence in an array of applicable actions. We choose only those actions that reduce the gap between the target value and the initial value.

#### 3.4.4.2 Evaluate Sentences

When we have a set of applicable actions, that is actions that we have just extracted because they contribute towards the goal as discussed above. It now remains to filter the applicable actions for those actions that satisfy the character traits and mood. This can be accomplished with several existing personality models. For instance the personality model created by Ball and Breese for Microsoft [5]. The applicable sentences should be qualified in some way by what type of personality they cater to and how they will be influenced by emotion. We mark each possible action with a range of traits and emotions that the action would be characteristic of. This means that if the actor bot is angry then the actor bot will be more likely to choose actions that indicate anger such as accusing someone. The actor bot's traits such as intelligence or arrogance will also make the actor bot more likely to choose a sentence that demonstrates this. For instance an intelligent actor bot will explain why a suspect has a motive.

Using BNs each sentence is evaluated and those that best match the characters traits and emotions are entered into a set of optimal action. From the set of optimal actions one action is chosen at random and executed.

### *3.4.5 Tests and Results*

Tests have been run in three ways, first is a test of computation speed, e.g. are the Bayesian algorithms sufficiently efficient to reach a decision within acceptable time limits, see Sect. 3.4.5.1. The second test is in the Second Life virtual reality, see Sect. 3.4.5.2. The third test is a web based game, see Sect. 3.4.5.3.

#### 3.4.5.1 Computation

To test our method we generated a Bayesian network from a script that encompasses an actor bot's belief about itself, its belief about other actor bots and its beliefs about other actor bots' beliefs about other actor bots. This network contains 3348 variables, one decision variable and one value variable, 3,638 arcs, 10,606 states and 58,610 parameters.

The tests were run on a laptop with Windows XP, Intel(R) Core(TM)2 Duo CPU 2.59 GHz, 3.00 GB RAM.

We tested two set-ups, one without Relevance Reasoning and one with Relevance Reasoning, see Table 3.2. When there is no Relevance reasoning applied it took the

**Table 3.2** Results

| *Queries* (%) | Without RR in seconds | With RR in seconds |
|---|---|---|
| 50 | 10–30 s | <1 |
| 30 | 30 s–3 min | <1 |
| 20 | >3 min | <1 |

actor bot on average 10–30 s to decide on an action and more than 20 % of the decisions took over 3 min.

When we use relevance reasoning it never takes more than a second for the actor bot to complete the whole process and decide on a sentence.

#### 3.4.5.2 Second Life

We need to take into account that the actor bots need to be animated in a virtual reality which needs a lot of resources and thus the AI computation may not take up all the time and resources available. We tested our implementation in the Second Life virtual reality [25] to see if the actor bots could perform at an acceptable speed when taking into account all the overhead of the virtual reality and animation that comes with it.

Second Life needs 25–33 % of the CPU time just for running the virtual reality display and no interaction is taking place. This means that the AI algorithm has reduced resources to manage the computations in.

The actor bots' responses should be a fluent reaction to user interaction. This means that response by the AI algorithms should not be so slow that the user becomes impatient and stops paying attention to the game.

In our implementation we needed to use a 5 s buffer to slow the actor bots down so that the user could keep up with what the actor bots were saying. Video clips showing a few of the test runs and explaining what is taking place can be found at http://youtu.be/HyVuHDSO1MI,http://youtu.be/dfFo8BNUap0, http://youtu.be/pfxz5RbmnFE.

#### 3.4.5.3 Web Game

To test whether the output of the engine are coherent sentences that can be used in drama based game we created a web demo prototype mystery game. As the game starts the player is told that a murder has been committed and he enters the murder scene where he can examine the body and determine the cause of death. Each initiated game has a randomly generated cause of death and the BN is used to generate clues on the body to be found by the user that are logically consistent with the cause of death. For example there will be marks around the neck if the


**Fig. 3.11** Screen shot of examining the body in the web game, note that what can be examined is also tailored by the BN depending on cause of death

victim was strangled and there may be strange odour if the victim was poisoned etc., see Fig. 3.11.

The player can then explore other rooms and interrogate suspects on their possible motives and attempt to determine which motive, if any, each suspect has. An example dialogue with two suspects shows how they are clearly different personalities, see excerpt below and in Fig. 3.12

Dialogue (Miss Jane Marple and Mr Bryan Eastley):

1. *Miss Jane Marple* - Did you lose a fortune?
2. *Mr Bryan Eastley* - what. Why? I have all my money
3. *Miss Jane Marple* - Did you have a reason to kill Harold?
4. *Mr Bryan Eastley* - Me? no..no reason!

Dialogue (Miss Jane Marple and Mr Cedric Crackenthorpe):

1. *Miss Jane Marple* - Did you lose a fortune?
2. *Mr Cedric Crackenthorpe* - No, I have all my money
3. *Miss Jane Marple* - Did you have a reason to kill Harold?
4. *Mr Cedric Crackenthorpe* - I assure you, I'm no murderer!

The game clearly demonstrated that the actor bots could act in character, e.g. show individual character traits and respond in a coherent manner to the questions asked by the player. More than 90 % of those who have played the game identify different character traits between the actor bots and find that the actors respond as they would expect murder mystery suspects in general to respond. The game is accessible online at http://www.weenrich.com.

The game is still quite simple but will be extended in the next few months to allow for more thorough testing with more detailed user evaluations.

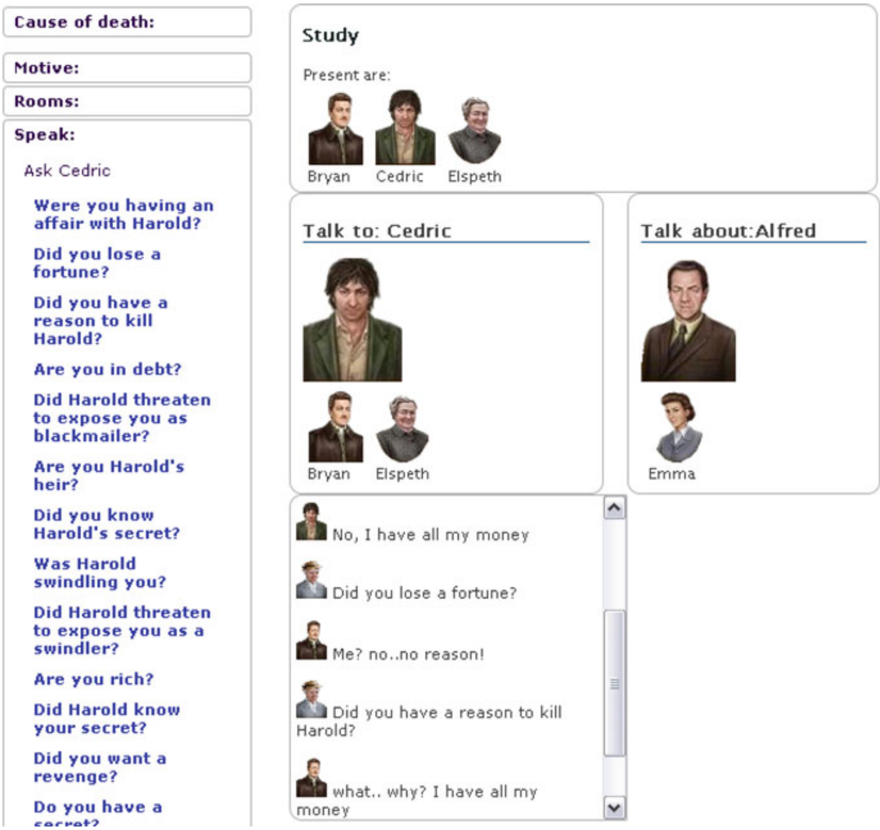



**Fig. 3.12** A screen shot of a dialogue with suspects in the web game

## *3.4.6 Complications*

The worst case scenario is that in order to update the target variables we need to update the whole network which is NP-hard. We reduce the chance of this by always having only a single target variable. Each target variable that is used in a query will need a certain portion of the BN to be updated given the input, the greater the number of target variables obviously the greater portion of the network needs to be updated and in the worst case the whole network would need to be updated. Instead of querying for many target variables simultaneously we query the net for each target variable and its inputs which allows us to reduce the size of the BN that needs to be updated. It is much more efficient to run multiple fast queries than a single large query that needs to update a large part of the BN.

Another complication is that if the output of the extracted subnet is needed in the big BN then the whole BN needs to be updated which means that our solution is not applicable for all instances, we still have not solved the problem of efficiently updating the whole network which will be necessary to some extent when the actor bots gain new information. Still the problem that we are looking at is to make the actor bot decide on a speech act and for this the actor bot does not need to update the whole BN.

## 3.5 Related Work

Impressive work in this area has been carried out at Southern California's Institute for Creative Technologies (ICT) and it is very well discussed in Chap. 4.

The most prominent and successful effort to take on complex social interactions is Façade [34]. Façade is an interactive drama where the player is asked to mitigate in a marital dispute between the players' friends. Short action sequences called beats control the sequence of events and these beats are explicitly pre-authored, with all actions within the beat being fully defined, and the actions of all roles being assigned to allow for multi-actor bot coordination [33]. Higher level goals for the characters are in these beats but the characters do retain autonomy in achievement of base-level goals and in performing actions such as facial expressions or personality moves [33]. The Façade characters are thus not fully autonomous bots, they are not acting as such but rather following directions from the drama manager and giving their actions a bit of personal character for believability.

FAtiMA (FearNot! affective mind architecture) is a character based emergent drama system [4, 38]. The drama emerges around character actions. The test base is FearNot! which is an anti bullying educational game to guide children in how to cope with bullying situations. The character agents are autonomous in interacting with the user by following a set of emotional reaction rules. These emotional reaction rules have pre and post condition and have appraisal values such as: desirability, desirability-for-other, and praiseworthiness [4, 21]. Using STRIPS-based partial-order continuous planner the characters evaluate the probability of success and importance of actions. Actions that are expected to generate the strongest emotional reaction are chosen.

FearNot! is the only system that has had proper user testing by an empirical study on 345 children, 172 male (49.9 %) and 173 female (50.1 %) between the ages of 8 and 11 [18]. The results showed that the children were able to empathise with the characters.

The Interactive Drama Architecture (IDA) [31] uses a director and a set characters that enact a fully structured story that is authored by a human author. The director in IDA gives direct commands to the characters, and the characters are semi-autonomous. They will act while they have no instructions from the director, for example drinking. They receive explicit directions from the director that have priority, for example 'perform dialogue #131 with John in the library and then run away to another room' [32]. This is very different from DED. In DED the agents do not get such explicit directions and they can decide to not comply with suggestions from the director by simply not choosing actions in schemas because they are not optimal for the drama development.

The story consists of plot points in a partially ordered graph using STRIPS with pre- and post-conditions. There is some variation in ordering of plot points, such as where a certain scene can take place. Still the main story is fixed and the user is modelled to enable the drama manager to subtly guide them through the storyworld using actions such as; deniers, causers, creations, shifters, hints.

Similar to this is the Mimesis [41, 51, 52] system which was created as part of the work of the Liquid Narrative Group. In Mimesis an attempt is made to give the user the illusion of complete freedom. The system evaluates each user actions and decides whether the user's action can be "accommodated" or must be "intervened" with. If the action can be accommodated a new plan needs to be constructed to achieve the story's goal. If accommodation is not possible the system must intervene with actions similar to IDA. Rather than pre-planning the DED system uses the BNs to evaluate in response to user action how the situation can best be developed further into a structured drama experience. This means that there is no call for these director actions because no plan is violated.

The BARDS system uses a Heuristic Search Planner (HSP) with RTA* to plan for emotional development in the characters, rather than for actions [39, 40]. The group use an ontology created by Gustave Flaubert as the basis for the planner. Flaubert's novel, *Madame Bovary* [11], provides the test scenario. The user using natural language makes comments which may cause characters to react emotionally and this will influence the story, for instance madam Bovary may feel guilt when she is flirting with her friend and is reminded of her children. The effect will vary depending on the characters' feelings and situation. This is a novel approach in that the user is the audience that can influence how the story unfolds. Similar to watching a movie and when the heroine is about to enter a murder infested room the audience can say: "No stop you will be killed!" and the heroine can choose to obey or not depending on her feelings. They have shown that planning can be used for this type of emotional evaluation sufficiently efficient to run dramas spanning a few minutes. The architecture is in all main areas significantly different from DED.

The virtual storyteller [44, 45] is a story telling system that creates stories from character agents interactions with an emphasize on emergent improvisational theatre. The stories are told by a narrative agent. There is no interaction between the character agents and the player; the player can only interact with the story teller.

Actor bots are specifically designed for RPGs. This is contrary to other related research for video games which aims to make bots for FPS, adventure games and strategy games because they do not involve complex interactions between player and NPCs [46].

The aims for Icarus [8], an automated FPS combat player, are the basic components that are needed for bots in games such as FPS if they are to be believable human. These include a hierarchical organization of knowledge with multiple levels of abstraction that provides a rich, human-like vocabulary and strategies. The bots use goal-directed and reactive behaviours and should support incremental learning and be able to use it later on in a believable fashion. Finally they should have the sufficient knowledge base to support complex reasoning. Icarus has a complex memory structure to support human like decision making based on well established cognitive theories. It does not accommodate emotion or characteristics though and its main goal is not interaction with the user in form of acting out an engaging drama like the actor bots.

This is the case with other similar bots for FPS and adventure games. Their aims are to play against the player rather than to be a companion or entertainment for the player as in Chaps. 7 and 8.

Imitating or learning from human players, as in Chap. 6 and 12, has seen some success in action based games such as in 3D boxing simulation game [35] and in FPS [16, 20, 53]. In the 3D boxing case based imitation showed a clustering of behaviour around that which the human player performed. There was a clear difference in the play of a bot imitating an experienced or weak player [16]. Use a Bayesian based probability function in a goal oriented bot. The bot imitates that actions of a human player and forms goal oriented strategies based on its observations [20]. Use pre-set probability distribution tables that the bot utilises with a Bayesian function. The bot can then learn new behaviours and playing styles by imitating other players. Finally [53] use actor bot modelling to learn good combat strategies for RPGs. All of these are successful in imitating combat actions of human players but do not contribute towards a more complex interactions as those handled by the actor bots.

## 3.6 Summary and Future Work

We have seen that current computer games are very good at computing character statistics and there are many quality games for power gamers and tacticians. Currently the game market has not been able to meet the demands of method actors and specialist or players that simply want a good flexible story. The reason is that current games tend to railroad players, forcing them to follow pre-authored stories.

It is the lack of a truly vicarious adventure that the Directed Emergent Drama (DED) engine aims to fill by facilitating the emergence of a structured drama from player and actors' interactions. In this chapter we shared how the DED engine facilitates similar experiences as provided by Pen and Paper Role-Playing Games, due to parallels in the fundamental approach to interactive drama.

Increasing basic role-playing concepts in Computer Based Role-Playing Games is very likely to increase player satisfaction by appealing to a larger audience and by increasing the degree of engagement and immersion. This remains to be tested with empirical user studies.

In this chapter, we presented real-time applicable algorithms based on relevance reasoning that are both efficient and scalable. Using Bayesian Networks (BNs) is a viable choice to represent the knowledge base and the characteristics of an intelligent agent. Since these agents are likely to need vast BNs, it is necessary for them to be able to decide on an action without updating a BN with possibly hundreds of thousands of variables. This we do by extracting the relevant subnet from the main BN and only updating the values in the subnet. Using d-separation and context we greatly reduce the size of the network that needs to be updated to compute inference of a target variable each time, effectively removing the exponential growth of BNs. Care needs to be taken to preserve d-separation where needed and to realise that the result needs to be propagated to the network when its influence on BN as a whole is needed.

We still have not worked out a way to update the main BN with information gained by the agent. This is a prominent problem that needs to be looked at. There are some possible solutions including hierarchical BNs and updating the BN in stages in separate threads using d-separation or clustering to divide it into ordered subnets that can be updated independently and the resulting values propagated to the neighbouring subnets.

In order to fully demonstrate scalability, the system needs to be applied to a larger domain. In particular, more content should be added so that the resulting BNs are both larger and more interconnected. Additionally adding greater complexity in the form of more traits and emotions will be good to test the bot's decision making. We expect that for a reasonably large drama an actor bot could have a knowledge base consisting of tens or hundreds of thousands of variables and it clearly remains to be solved how such a large BN would be authored by a non-programmer.

As future work it will be interesting to include psychosocial behaviours and deeper personality and emotional reaction such as described in Chap. 2. The framework defined there would go very well with a Bayesian Network.

Additionally it would be constructive to explore how Recursive Modelling Method (RMM), as described in [15], can aid in giving the agents a more complete theory of mind without needing to go into full game theory and the subsequent search for Nash Equilibrium [36].

The engine is equally able to operate on physical actions as on speech actions and in our future experiments we will add this.

## References

1. Arinbjarnar, M., Kudenko, D.: Schemas in directed emergent drama. In: Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling (ICIDS08), Erfurt (2008)
2. Arinbjarnar, M., Kudenko, D.: Duality of actor and character goals in virtual drama. In: Proceedings of the 9th International Conference on Intelligent Virtual Agents, Amsterdam (2009)
3. Arinbjarnar, M., Kudenko, D.: Bayesian networks: real-time applicable decision mechanisms for intelligent agents in interactive drama. In: Computational Intelligence and Games (CIG), Copenhagen, Aug 2010
4. Aylett, V., Dias, J., Paiva, A.: An affectively driven planner for synthetic characters. In: Proceedings of International Conference on Automated Planning and Scheduling (ICAPS06), UK (2006)
5. Ball, G., Breese, J.: Relating personality and behavior: posture and gestures. In: Paiva, A. (ed.) Affective Interactions, pp. 196–203. Springer, Berlin (2000)
6. Bioware. Newerwinter nights. http://nwn.bioware.com/ (2002)

7. Bohus, D., Horvitz, E.: Dialog in the open world: platform and applications. In: Proceedings of the 2009 international conference on Multimodal interfaces (ICMI-MLMI '09), pp. 31–38. ACM, New York (2009)
8. Choi, D., Konik, T., Nejati, N., Park, C., Langley, P.: A believable agent for first-person shooter games. In: Proceedings of the Third Annual Artificial Intelligence and Interactive Digital Entertainment Conference, pp. 71–73. AAAI Press, Stanford (2007)
9. Sony Computer Entertainment. Heavy Rain, http://www.heavyrainps3.com/ (2009)
10. FASA. Star trek: The role playing game (1982)
11. Flaubert, G.: Madame Bovary. France, Paris (1856)
12. Freytag, G.: Technique of the Drama. Benjamin Blom, New York (1863)
13. Goodman Games. Etherscope, http://www.goodman-games.com/etherscope.html (2006)
14. Glymour, C., Danks, D.: Reasons as causes in bayesian epistemology. J. Philos. **104**(9), 464–474 (2007)
15. Gmytrasiewicz, P.J., Durfee, E.H.: A rigorous, operational formalization of recursive modeling. In: International Conference on Multiagent Systems, pp. 125–132 (1995)
16. Gorman, B., Thurau, C., Bauckhage, C., Humphrys, M.: Believability testing and bayesian imitation in interactive computer games. In: From Animals to Animats 9, vol. 4095, pp. 655–666. Springer, Berlin (2006)
17. Gygax, G.: Dungeon Module B2: The Keep on the Borderlands—Introductory Module for Character Levels 1–3. Berkeley Top Line Distributing, Berkeley (1980)
18. Hall, L., Woods, S., Aylett, R., Newall, L., Paiva, A.: Achieving empathic engagement through affective interaction with synthetic characters. In: Proceedings of the International Conference on Affective Computing and Intelligent Interfaces, pp. 731–738. Springer, Berlin (2005) (LNCS 3784)
19. Horvitz, E., Paek, T.: A computational architecture for conversation. In: Proceedings of the Seventh International Conference on User Modeling (UM'99), pp. 201–210. Springer, New York (1999)
20. Le Hy, R., Arrigoni, A., Bessiere, P., Lebeltel, O: Teaching bayesian behaviours to video game characters. Robotics and Autonomous Systems (Elsevier) Special issue: Robot Learning from Demonstration, **47**, 177–185 (2004)
21. Dias, J., Paiva, A.: Feeling and Reasoning: A Computational Model for Emotional Characters. In: 12th Portuguese Conference on Artificial Intelligence (EPIA 2005), pp. 127–140. Springer, Berlin (2005)
22. Jensen, F. V.: Bayesian Networks and Decision Graphs. Springer, Berlin (2001)
23. Juul, J.: Games telling stories? Int. J. Comput. Game Res. **1**(1) (2001)
24. Koller, D., Pfeffer, A.: Object-oriented bayesian networks. In: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, pp. 302–313. San Francisco (1997)
25. Linden Lab. Second Life, http://secondlife.com/, June (2008)
26. Lagnado, D.A., Waldmann, M.R., Hagmayer, Y., Sloman, S.A.: Beyond covariation: cues to causal structure. In: Gopnik, A., Schultz, L. (eds.) Causal Learning: Psychology, Philosophy, and Computation, 1st edn. Oxford University Press, USA, (2007)
27. Laurel, B.: Computers as Theater. Addison-Wesley Publishing Company, Boston (1993)
28. Laws, R.D.: Robin's Laws of Good Game Mastering. Steve Jackson Games, Austin (2002)
29. Lin, Y., Druzdzel, M.: Computational advantages of relevance reasoning in bayesian belief networks. In: Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), pp. 342–35. Morgan Kaufmann, San Francisco (1997)
30. Bethesda Softworks LLC. Fallout 3. http://fallout.bethsoft.com/index.html (2002)
31. Magerko, B.: Story representation and interactive drama. In: Proceedings of the Artificial Intelligence and Interactive Digital Entertainment conference (AIIDE) (2005)
32. Magerko, B.: Player modeling in the interactive drama architecture. Ph.D. thesis, The Department of Computer Science and Engineering, University of Michigan (2006)
33. Mateas, M.: Interactive drama, art, and artificial intelligence. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh. Technical Report CMU-CS-02-206 (2002)

34. Mateas, M., Stern, A.: Build it to understand it: ludology meets narratoloy in game design space. In: Proceedings of the Digital Interactive Games Research Association Conference, Vancouver. Included in the Selected Papers volume, June 2005
35. Mozgovoy, M., Umarov, I.: Building a believable agent for a 3D boxing simulation game. In: 2010 Second International Conference on Computer Research and Development, pp. 46–50. May 2010
36. Nash, J.: Non-Cooperative Games. Ann. Math. **54**(2), 286–295 (1951)
37. Oxford, English: and Dictionary. Oxford University Press, Oxford English Dictionary (2007)
38. Paiva, A., Dias, J., Sobral, D., Ayllet, R., Sobreperez, P., Woods, S., Zoll, C., Hall, L.: Caring for agents and agents that care: building emphatic relations with synthetic agents. In: Proceedings of the Autonomous Agents and Multi-agent Systems (AAMAS), ACM Press, UK (2004)
39. Pizzi, D., Cavazza, M.: Affective storytelling based on characters' feelings. In: Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies, Arlington, Nov 2007
40. Pizzi, D., Charles, F., Lugrin, J., Cavazza, M.: Interactive storytelling with literary feelings, In: Proceedings of the Second International Conference on Affective Computing and Intelligent Interaction (ACII), Lisbon, Sept 2007
41. Riedl, M.O., Saretto, C., Young, R.M.: Managing interaction between users and agents in a multi-agent storytelling environment. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2003)
42. Schiesel, S.: Dungeon masters in cyberspace. The New York Times (2006)
43. Schiesel, S.: Gary Gygax, game pioneer, dies at 69. The New York Times, 157(54240) (2008)
44. Swartjes, I., Vromen, J.: Emergent story generation: lessons from improvisational theater. In: Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies, Arlington, Nov 2007
45. Swartjes, I., Vromen, J., Bloom, N.: Narrative inspiration: using case based problem solving to support emergent story generation. In: Proceedings of the International Joint Workshop on Computational Creativity, Goldsmiths, University of London, June 2007
46. Tencé, F., Buche, C., De Loor, P., Marc, O.: The challenge of believability in video games: definitions, agents models and imitation learning. CoRR, abs/1009.0451 (2010)
47. Tenenbaum, J.B., Griffiths, T.L.: Structure learning in human causal induction. Adv. Neural Inf. Process. Syst. **13**, 59–65 (2001)
48. Thue, D., Bulitko, V., Spetch, M.: Passage: a demonstration of player modelling in interactive storytelling. In: The Fourth Conference on Artificial Intelligence and Interactive Digital Entertainment, AAAI Press, Palo Alto (2008)
49. Ubisoft. Assassin's Creed II, http://assassinscreed.uk.ubi.com/assassins-creed-2/ (2009)
50. GSC Game World. S.T.A.L.K.E.R., http://www.stalker-game.com/ (2009)
51. Young, R.M., Riedl, M.: Towards an architecture for intelligent control of narrative in interactive virtual worlds. In: Proceedings of the International Conference on Intelligent User Interfaces, Jan 2003
52. Young, R.M., Riedl, M.O., Branly, M., Jhala, A., Martin, R.J., Saretto, C.J.: An architecture for integrating plan-based behavior generation with interactive game environments. J. Game Dev. Vol. 1 (2004)
53. Zhao, R., Szafron, D.: Learning character behaviors using agent modeling in games. In: Proceedings of the Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-09), pp. 179–185, Oct 2009